

Automating Frequency Tables with ClinXport SAS Macro Package

Loizos Nicolaou, ClinBAY, Limassol, Cyprus
Julien Vanwinsberghe, ClinBAY, Genappe, Belgium

ABSTRACT

The use of standardized clinical data formats such as SDTM and ADaM from CDISC is constantly growing in the clinical research field. Therefore, automating clinical data processing and analysis becomes increasingly achievable. At ClinBAY, we have developed and used a software called ClinXport®, to report clinical data for more than 150 studies so far. This tool consists of a Microsoft Excel graphical interface and a package of SAS® macros that enable creation of most standard TFLs, without having to write a single line of SAS code. In this article, we will present one of the ClinXport SAS macros that automates the creation of frequency tables, mostly used for reporting adverse events and demographics data. After presenting the global behavior of the reporting system, we will explain in detail the functional process of the macro, so that the user can implement a similar approach in his production environment.

INTRODUCTION

Owing to the progressive increase in standardization of clinical data formats, clinical data processing and analysis are becoming increasingly amenable to automation. The current paper describes and explains a SAS macro package (called REP_AEV), which enables automation of frequency tables production. A thorough explanation on how to exploit the tools of a software developed and used by ClinBAY called ClinXport is also given, in order to build frequency tables in a SAS reporting environment. It is important to note that the package described and explained in the current paper is suitable for any SAS user. On one hand, the friendly interface allows beginners to build or modify tables in a simple way. On the other hand, a number of tools give the flexibility required for producing complex tables, to more experienced users and specialists.

A GENERAL BRIEFING OF HOW CLINXPORT WORKS

ClinXport consists of an Excel file, and a library of SAS macros driving all reports in a study. The Excel file contains the following:

- A ClinXport add-in toolbar (Figure 1), that provides several functionalities such as defining meta-data, exporting data to SAS, executing SAS for generating outputs, opening programs and outputs etc.
- A STUDY sheet, containing study-specific information, common in all outputs (e.g. study title).
- A DOMAIN sheet (Figure 2), containing the list of outputs, commonly known as Tables, Figures and Listings (TFLs), and associated output information such as: output title, output type, input data location, output location, and a link to template sheet, containing information required to produce a specific output.
- A FORMATS sheet (Figure 3), containing formatted values that can be applied to SAS variables.
- A template sheet (Figure 4), containing output-specific information (e.g. title, footnote, table columns), in order to produce the desired output. The STUDY and DOMAIN sheets, but mainly the template sheet (also known as shell), enable the user to control and customize output layout.

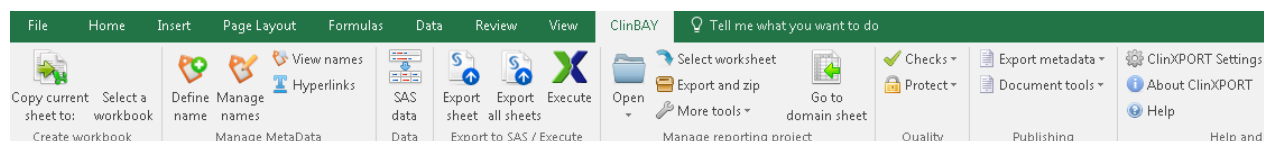


Figure 1: Screenshot of the ClinXport toolbar.

Template Name	Output Type (Table, Listing, Figure)	Output number	Table title	Population	Output filename (without extension)	Dataset Name	Program Name (without .sas extension)
AE01	Table	1.1	Treatment emergent Adverse events, regardless of study treatment relationship	Safety analysis set	t1_01	ADAE	rep_aev
AE01	Table	1.2	Treatment emergent Adverse events related to study treatment	Safety analysis set	t1_02	ADAE	rep_aev

Figure 2: Screenshot of the DOMAIN sheet.

PhUSE 2016

	A	B	C	D	E
1	Format name	Format type (VALUE, INVALUE, PICTURE)	Value type (num or char)	In Value	Out Value
56	TRTAN	value	char	01	Drug A
57				02	Drug B
58				999	Total
59	SEV	value	num	1	Mild
60				2	Moderate
61				3	Severe

Figure 3: Screenshot of the FORMATS sheet.

In general, ClinXport works by using SAS macros that read and identify meta-data from these Excel worksheets, which are consequently used for creating TFLs. The ClinXport toolbar is used for managing the macro variable ranges required for TFL reporting, executing SAS, visualizing data and outputs, and more.

HOW DOES THE REP_AEV MACRO WORK

After having discussed the global behavior of the ClinXport software, we will get into more detail about the process of using a specific macro package (REP_AEV) for producing frequency tables, without having to write a single line of SAS code. It is important to note that a similar methodology can be applied to any reporting environment using SAS macros, and it is not restricted to use within ClinXport. Screenshots will be used in this section, to demonstrate both the input template sheet containing the output-specific information, as well as the final output. An example of a common output for clinical trials, the incidence of adverse events by System Organ Class (SOC), Preferred Term (PT), severity and treatment, will be used for demonstration purposes in the current paper.

The REP_AEV macro can be called under the “Program Name” column of the DOMAIN worksheet (Figure 2). This macro consists of the following two components: (1) The specific macro features that are used for calculating the frequency statistics, and (2) the general ClinXport features that are used to create table reports using the PROC REPORT procedure in SAS.

THE TEMPLATE SHEET

	A	B	C	D	E	F	G	
1	&studynumber JUST_L							title
2	&outputtype. &outputnumber. (&pageofy.)							
3	&outputtitle.							
4	by system organ class, preferred term and maximum severity							
5	Analysis Set : &analysisset.							
6								
7	System organ class#	Severity	Drug A	Drug B	Total		header	
8	%str()Preferred term		xx mg	xx mg	xx mg			
9			n(%)	n(%)	n(%)			
10	Number of subjects with AEs	Mild	xx (xx.x)	xx (xx.x)	xx (xx.x)			
11		Moderate	xx (xx.x)	xx (xx.x)	xx (xx.x)			
12		Severe	xx (xx.x)	xx (xx.x)	xx (xx.x)			
13	Cardiac disorders	Mild	xx (xx.x)	xx (xx.x)	xx (xx.x)			
14		Moderate	xx (xx.x)	xx (xx.x)	xx (xx.x)			
15		Severe	xx (xx.x)	xx (xx.x)	xx (xx.x)			
16	Sinus bradycardia	Mild	xx (xx.x)	xx (xx.x)	xx (xx.x)			
17		Moderate	xx (xx.x)	xx (xx.x)	xx (xx.x)			
18		Severe	xx (xx.x)	xx (xx.x)	xx (xx.x)		footnote	
19	Program: &programpath.\&programname..sas %sysfunc(date(),date3.):%sysfunc(time(),time5.) &programmode.							
20								
21								
22								
23	AEBODSYS if compress(AEDECOD)eq'TOTAL'the AESEVN		drug_lab=cc value4				_colvar	
24	group		group flow	across flow			_colopt	
25	order=data	order=data	order=data	order=data				
26	cluster1	format=sev.	totalb	treatvar	group			
27	total	format=\$nowaes. cluster1total	max {12 3}					
28	Subject analysis dataset		ASSV				macro specific variables	
29	Subject analysis subset		SAFFN=1 and TRTAN>2					
30	Subject variable		USUBJID					
31	Variable sequence		AEBODSYS AEDECOD AESEVN TRTAN					
32	Sort by		abc					
33	Preprocess							

PhUSE 2016

recall that the ClinXport toolbar can be used to define and manage these regions. Anything outside the defined macro variable regions will not impact the created output. It is possible to use a similar approach outside the ClinXport environment, by defining the same information in macro variables within a SAS program, or in metadata, such as CSV files.

TEMPLATE MACRO REGIONS

Information on the different template macro regions is given in the descriptions below:

TITLE: Contains study number information from the STUDY sheet, as well output specific information such as title and output number, found in the domain row of each output (Figure 2).

HEADER: Used to create column headers. Just like any other macro region, macro variables from STUDY and DOMAIN sheets or simple text, can be used for defining header names (Figure 4). The only exception is in the case of any across variables, headers are defined as per the variable categories, rather than the information in the defined range.

_COLVAR: Contains the names of the variables that need to be included in the report. Cells of the _colvar region above (Figure 4), contain the corresponding variables for SOC, PT, severity and treatment. SAS code can be used within the cells, in order to perform data manipulation or derivations for presentation purposes, thus allowing for further flexibility.

_COLOPT: Contains the options that can be applied to the corresponding report variables in the _colvar region. There are two kinds of options available for use; the proc report options and rep_aev macro specific options. The proc report options such as group, across, order=, format=, noprint, id, etc. are used in the same way they would have been used in the “define” statement of the PROC REPORT procedure. The macro-specific options will be discussed in more detail in the section MACRO OPTIONS.

MACRO-SPECIFIC VARIABLES

These are required macro variables that are mostly specific to the REP_AEV macro, and include:

SUBJ_ANALYS_DATA: Contains the name of the analysis dataset, which includes the subjects to be used as the reference group for deriving frequency percentages for each treatment group (e.g. ASSV).

SUBJ_ANALYS_SET: Can be used to include any subset conditions that need to be applied on the subject analysis dataset defined above (e.g. SAFFN=1 and TRTAN>2).

SUBJVAR: Requires the subject identity variable (e.g. USUBJID).

GROUPVAR: Requires the sequence of variables that will be used for reporting. In Figure 4 above, the four variables correspond to SOC, PT, severity and treatment group (e.g. AEBODSYS AEDECOD AESEVN TRTAN).

SORTBY: This specifies the desired rule of sorting categories in the report. Options can be either ABC or FREQ corresponding to sorting alphabetically or by descending frequency in the “total” group respectively, given that “total” groups have been created.

PREPROC (optional): SAS code can be included in this optional region, such as “if” statements, to perform data preprocessing.

MACRO OPTIONS

As mentioned above, apart from the reporting options, there exist a number of powerful macro-specific options that enable the user to create frequency tables with user-desired specifications. A description of these options along with a depicted example are provided below:

- **Cluster i :** Can be used to determine variables with dependent responses, forming part of cluster group i .
- **TotalT/TotalB:** Can be used to create total category rows/columns on top or bottom respectively for any variable.
- **Max/Min:** These can apply on numeric variables and give the possibility of counting a subject with occurrences of an event in different grades in the maximum /minimum grade only.
- **{1, 2 ... n}:** Can be used to determine the numerical categories that can be reported in the table, even if no occurrences were observed for some of the numerical categories.

Let us consider an example below to see how these options are used within the _colopt defined region (Figure 5) and the resulting executed table.

PhUSE 2016

CLUSTER1: This option is inserted in the first two cells of the `_colopt` region corresponding to columns for SOC and PT, in order to create a table with dependent responses associated with these two columns. Had this option not been available, all possible PT categories would have appeared within each distinct SOC group.

TOTALT: The first two cells of the `_colopt` region also contain the `totalt` option, which creates a total group for the SOC, and each distinct group of PTs within the same SOC. The “Number of subjects with AEs” appearing in the table below, is the formatted value corresponding to the created total group. The `totalt` option in the 2nd cell, creates a total row for each group of PTs within the same SOC category. The categories in the orange boxes below (Figure 5) comprise an example of the total rows created for two groups of PTs belonging to two distinct SOC. As shown, the SOC itself is used as the label for each of those total rows. The feature of using SAS code within the cells of the `_colvar` regions enables the user to achieve greater flexibility in presentation. Below we can see the 2nd cell of the `_colvar` region expanded.

```
if compress(AEDECOD) eq 'TOTAL' then VAR2=propcase(AEBODSYS);  
else VAR2=propcase('      '||AEDECOD);
```

In essence, what this does, is to redefine the total row of each group of PTs as the SOC itself. Otherwise, non-total PTs are indented, as shown in the example below (Figure 5). Notice that the information from the first two columns of SOC and PTs is concatenated, and presented in the second column only. SOC categories are not reported in the initial column, as the `noprint` option is inserted in the 1st cell of `_colvar`.

Another example of using SAS code is found within the 4th cell of `_colvar` region shown expanded below. This across variable creates a label for the treatment headers including the drug name and number of subjects within this treatment group, as shown in the purple box of Figure 5.

```
drug_lab=compbl(put(TRTAN,TRTAN.)||'#N='||compress(put(N0,5.0)));
```

TREATVAR: The number of subjects in each treatment group in the code above, `N0`, is created by the macro and used as the denominator for deriving percentages in the adverse event table (Figure 5). The `treatvar` option inserted in the 4th cell of `_colopt` region below, is an option that needs to be used, in order to specify which variable will be used as the classification variable for calculating the number of subjects `N0`, explained above.

MAX: The option specified under the `_colopt` cell corresponding to severity variable, results in counting a subject with occurrences of an event in different grades in the higher grade only. In the blue box in Figure 5, we can see a total of 11 subjects with mild event(s) and 2 subjects with moderate event(s). It is possible that these 2 subjects also experienced mild events, however, they are only counted in the moderate group due to the use of the `max` option.

{1, 2, ..., i}: Numbers within curly brackets determine the skeleton of reported numerical values of the variable. By default, the macro will report only the numerical categories observed. The user is capable of reporting additional numerical categories, by simply typing those categories within curly brackets. The example below (Figure 5) shows how the user was able to include the “severe” category in the report, even though no severe events were observed. It should be noted that severity is a numeric variable, with mild moderate and severe being the corresponding formatted values (Figure 3).

PhUSE 2016

```

23 AEBOBSYS if compress(AEDECOD) eq 'TOTAL' the AESEVN drug_lab=cc value4
group
order=data
noprint
cluster1 group flow order=data id
total format=$nowaes. cluster1 total
group flow across flow
order=data order=data
format=sev. totalb
max {1 2 3} treatvar group

```



Demo study

Table 1.1 (Page 1 of 5)
Treatment emergent Adverse events, regardless of study treatment relationship
by system organ class, preferred term and maximum severity
Analysis Set : Safety analysis set

System organ class Preferred term		Drug A 10 mg N=22		Drug B 50 mg N=22		Total N=22	
Number of subjects with AEs		Mild	8 (36.4)	6 (27.3)	11 (50.0)		
		Moderate	2 (9.1)	1 (4.5)	2 (9.1)		
		Severe	0	0	0		
Blood And Lymphatic System Disorders		Mild	0	0	0		
		Moderate	1 (4.5)	0	1 (4.5)		
		Severe	0	0	0		
Eosinophilia		Mild	0	0	0		
		Moderate	1 (4.5)	0	1 (4.5)		
		Severe	0	0	0		
Gastrointestinal Disorders		Mild	2 (9.1)	2 (9.1)	3 (13.6)		
		Moderate	0	0	0		
		Severe	0	0	0		
Abdominal Pain		Mild	0	1 (4.5)	1 (4.5)		
		Moderate	0	0	0		
		Severe	0	0	0		
Dry Mouth		Mild	2 (9.1)	0	2 (9.1)		
		Moderate	0	0	0		
		Severe	0	0	0		
Toothache		Mild	0	1 (4.5)	1 (4.5)		
		Moderate	0	0	0		
		Severe	0	0	0		

Program: N:\Software\ClinXPORT\releases\v2\v2.2\SAS\BUM\compiled_x32\rep_aev.sas 02AUG2016:11:14

Figure 5: _colvar and _colopt regions / Adverse event table.

OPERATIONAL STEPS OF REP_AEV MACRO

At this point we will briefly summarize the operational steps of the REP_AEV macro, so that users can apply a similar methodology to create frequency tables in their own reporting environment.

1. Import of SAS data and Excel metadata.
2. Data processing prior to counting frequencies, such as creating total groups.
3. Frequency count calculation using PROC SQL, taking into account max/min options.
4. Creation of report skeleton, adjusting for clustered variables and numerical unobserved categories that should be included in the report.
5. Merging of skeleton dataset (created in step 4) and dataset of frequency results (created in step 3) creates the final dataset, which will be subsequently used for reporting.
6. Production of frequency table using PROC REPORT.

A challenging part of the macro that is worth elaborating on, is the handling of the interaction of the total/totalb and max/min options on numeric variables. In order to create a total group for a numeric variable, the macro simply duplicates the observations in the dataset, assigning a default value of "999" as the new numerical response. These observations are then

PhUSE 2016

counted separately to give frequency results for the total group. Without any processing, the max option would have prevented the reporting of actual severity categories (e.g. mild, moderate and severe), as it would have identified the “999” records as the ones of maximum severity for any given subject and AE category. Similarly, the use of a min option would have automatically prevented the reporting of total groups.

The macro overcomes the abovementioned challenge, by splitting the dataset including total records (created at step 2 of operational steps) into 2^n datasets, with n being the number of numeric variables in the AE report, and 2 being the level of groups to be considered separately when performing the max/min procedure. These two levels would be the total and non-total groups. The example used in this paper is a simple case with just one numeric variable. Hence, after step 2 explained above, the macro will split the dataset into 2^1 datasets, separating total and non-total records. It will then use a PROC SQL procedure on the two distinct datasets, to count the number of subjects with occurrences of an event in different grades in the maximum grade only; separately for the total and non-total categories. The macro is dynamic and can apply the same process, should more numeric variables with total and max/min options be required in the report.

GOING FURTHER

Following on from work in the current paper, if a table is too complex to produce with the automated procedure, SAS custom programs can be used instead. The REP_AEV macro can be called in a SAS program, enabling performance of complex manipulations before and after execution of the macro. SAS programs are referenced in the Excel sheet and are executed automatically.

CONCLUSION

The paper shows how the new ClinXport macro, REP_AEV, enables the user to create complex frequency tables without using SAS code. The ease of reusing or adapting existing templates, leads to simplification of TFL reporting, and thus a significant gain in productivity. Additional ClinXport SAS macros can be used to produce standard TFLs in a very similar manner. Importantly, this methodology is not restricted to the ClinXport environment, and can be applied by SAS users in their own reporting environment. Development of such macros, will improve efficiency and contribute towards reducing time in the process of clinical trial reporting.

REFERENCES

ClinXport a software to streamline the analysis and reporting of clinical trials with SAS.

<http://www.phusewiki.org/docs/2011%20Papers/SD10%20paper.pdf>

ACKNOWLEDGMENTS

We would like to thank Maria Pieri for her constructive suggestions that improved the paper.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Loizos Nicolaou
ClinBAY
Office 401, Vanezis Business center, 171 Arch Makariou III Avenue
3027 Limassol
Cyprus
Work Phone: +32 67 70 90 80
Email: loizos@clinbay.com
Web: www.clinbay.com

Brand and product names are trademarks of their respective companies.